Proceedings of the 16th International Conference on Automation & Computing, University of Birmingham, Birmingham, UK, 11 September 2010

Video Feature Representation and 3D Curve-based Event Matching

Jing Wang*, Michael O'Grady, Qian Xu, Zhijie Xu
Department of Informatics, School of Computing and Engineering
University of Huddersfield
Queensgate, Huddersfield HD1 3DH, United Kingdom
j.wang2@hud.ac.uk, m.ogrady@hud.ac.uk, q.xu@hud.ac.uk, z.xu@hud.ac.uk

***Abstract*—This paper highlights the progress of the research programme for investigating spatio-temporal volume based video event detection. The main research aim is to devise innovative and efficient volume-based video content analysis techniques and systems. To tackle this challenge, this paper fouces on the subjects of volume based feature analysis and event template matching. Corresponding system prototype design and experimental result analysis have also been presented in this report. The experiment result has clearly demonstrated the advantages of the volume based event detection methodology in terms of its richness of temporal-related information and the potentials for identifying complex activities. As an important application, the difficulties in representing/extracting dynamic features from randomly ordered stacks of 2D snapshots can now be readily denoted as voxels for analysis in a 3D volumetric space.**

Keywords-event detection; volume processing; video processing; feature extraction

I. INTRODUCTION

Spatio-Temporal Volume (STV) data structure was firstly introduced by Aldelson and Bergen [1] in 1985, to emphasize the temporal-related features embedded in video data. This research has adopted the STV idealism for video-based dynamic event detection. To facilitate the quick comprehension of basic concepts of STV-based processing and the motivation of this project, Fig.1 shows an STV model being represented as a volumetric object in a 3D coordinate system denoted by x, y and t (time-dimension) axes. The STV model is composed of a stack of video frames assembled by 2D arrays of pixels in the time order. In this structure, an individual frame is represented by the pixel values corresponding to the x-y coordinate, while the dynamic information mainly preserved and represented by the segmentation and the navigation path of certain sets of pixels. To integrate the spatial and temporal information in a unified data structure for processing and analysis, each fundamental element inside of a STV model is referred at here as a voxel (volume pixel), a concept inherited from computer graphics and visualization research.

Compared with the traditional frame-by-frame-based video analysis, a significant advantage of STV is its distinct ability to provide direct mathematical descriptions for dynamic features extracted from a section of video footage. The information can then be further processed to identify a pre-defined video event. In this research, the feature collections - known as "stick-model" - are highlighted as human-like shapes, composing head, arms and legs as, illustrated in Fig.2. The coordinates of those voxels are considered essential features that need to be extracted from each frame. The trajectories and envelopes of those voxels can then be used to form the slimmed version of the STV models to represent dynamics and movement of a human body. The result of this process can then be used for matching against the pre-constructed event templates in the 3D volumetric space.

This project has also built upon the famous Rogez's stick model [2] for constructing event templates in the STV feature space, which, in direct observation, are similar to different forms of 3D curves as shown in the Fig.3. Dynamic events and human actions can be modeled this way and being represented by various geometrical and topological structures.

After establishing event feature points in a 3D volume space, the task of event detection problem is converted into matching the pre-determined event templates with the extracted curve patterns. However, preliminary experiments performed in this research show that curves may share identical geometrical features and are difficult to separate using conventional methods. To solve this pattern analysis problem, research has also been carried out to devise a more robust method in two inter-related steps: curve geometries analysis and "String"-based 3D curve matching. Fig.4 illustrates the entire operational pipeline of this improved method.

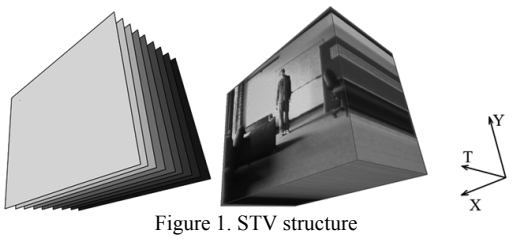

Figure 1. STV structure

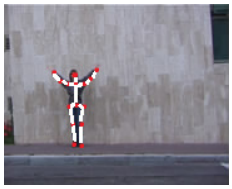
Figure 2. Human stick-model

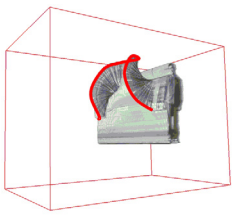
Figure 3. Waving event curve in STV feature space

* Presenter of this paper

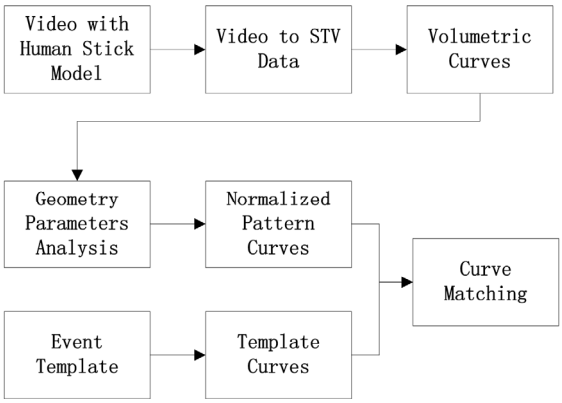

Figure 4. STV curve-based video event detection system pipeline

This paper is organized in the following order: Section II highlights the curve geometry analysis methods employed in this research. Section III provides details on 3D curve matching algorithms. Section IV forced on the development operations and the final experiment result analysis. Summaries and future works are discussed in section V.

II. CURVE PARAMETERS ANALYSIS

In an STV space, different feature curves display distinctive geometric distributions. For example, static features are often shown as straight lines which might be perpendicular to the XY plan in the volume coordinate system shown in Fig.5.A. Through investigating the alignment and slope angles of those lines and the periodical characteristics of different waves, the category of a human event can be identified.

By introducing the Least Absolute Residual [3] (LAR) linear fitting methods and Mean Absolute Error [4] (MAE) statistical operations, it is clear only still or uniform motion of the target object will show the distribution of feature points as straight lines. This inherent feature can facilitate the separation of liner and non-linear movements from dynamic "points". For example, as shown in the Fig.5.B and 5.D, the walking and bending events can be readily separated by analyzing the linearity of the head movement. The slope angle of a straight line against the XY plan in the 3D volume space reflects the speed of a linear motion, which can be calculated by using the linear fitting algorithm. This parameter can be used for setting up thresholds for differentiating linear movements such as stand still, walking and running. Fig.5.A, B and C show the feature curves for different head states.

Some events such as waving and walking are repetitive human events, which are represented as periodic waves in a 3D space. A periodic event is more difficult to model in the feature space since the unknown factors for template matching because of periodicity and frequency. Based on the autocorrelation method by Parr [5], the periodic parameters can be approximated by evaluating curves and peak values as shown in the example illustrated in Fig.6.

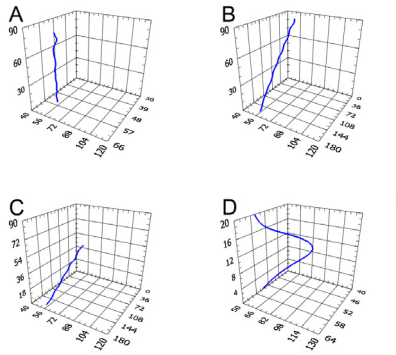

Figure 5. Head curves of the stand still, walking, running and bending event

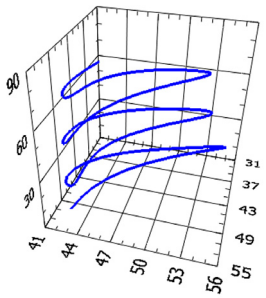
Figure 6. The curve of a waving event

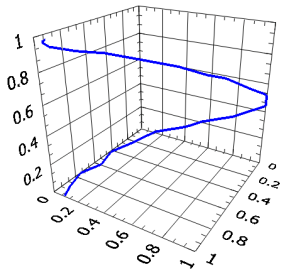
Figure 7. Normalized curve of the same waving event

To speed up the event detection process, in this research, template curves for periodic events have covered only a single periodical cycle, which can be readily expended for any particular application requirements. The periodical characteristics can be normalized in the Euclidean space by controlling the location of the start point and the span of a period. Fig.7 shows one normalized waving event.

III. 3D CURVE MATCHING FOR VOLUME-BASED EVENT DETECTION

In the 3D feature volume space, event templates and patterns are defined as 3D curves. Classification and recognition of these human events required curve-based template matching. During this research, the pattern recognition is mainly achieved by measuring the difference between the filtered and constructed event patterns and the pre-defined event templates. This section discusses an innovative 3D curve matching approach developed in the program.

In 1993, Bunke [6] devised a "Strings Edit" operation for 2D shape recognition based on Wagner and Fischer's earlier work [7]. The method accumulates differences of two 2D curves by matching each element of the two converted strings. The total difference is generated by counting the cost on relevant editing operations which change the unknown string into the template string. The "edit complexity" and "cost" can be quantified by a non-negative function $c(e)$, where e denotes three basic String Edit operations, known as insert, delete and substitute. Given two strings $\mathbf{X}=x_1, x_2, \ldots, x_n$ and $\mathbf{Y}=y_1, y_2, \ldots, y_m$, the non-negative differences $d(\mathbf{X},\mathbf{Y})$ is defined by the minimum cost of editing $\mathbf{X}$ into $\mathbf{Y}$. if $d(\mathbf{X},\mathbf{Y})=0$, it means the two strings are exactly the same. More differences between two strings means bigger $d(\mathbf{X},\mathbf{Y})$. Thresholding this value can control the matching outcome.

This string-based sequencing and editing operations can be adopted for any pattern matching problems if (and only if) the pattern can be completely mapped as a series of feature. For example, a closed shape can be described by coordinates of its contour in Euclidean space. The serial features of these coordinates denote a distinctive curve in pattern space. Following subsections provide details on the investigation into extending the technique for 3D open curve matching.

A. Curvature and Torsion of Curve

Dynamic human features are represented by feature curves in the STV structure. This simplification enables the conversion of video events into STV feature volumes for matching and analysis. Comparing the coordinates between the template curve and an unknown pattern curve is the most popular method. However, in human event detection applications, it is difficult to compares two curves without taking into account their 3D natures [8]. This problem has been overcome in this project by introducing the curvature and torsion parameters which are often related to analytical geometry. Movement events can be categorized by comparison of their curvature and torsion pair values with those from the template.

For example, if a defined curve, $\mathbf{r}$, in the Euclidean space can be defined as,

$$\mathbf{r}(t)=\begin{bmatrix}x(t) & y(t) & z(t)\end{bmatrix}, \tag{1}$$

which represents a series of positions (vectors) by the changes of t. Moreover, it can represent a curve with arc length s which is defined as,

$$s(t)=\int_0^t \left\|\mathbf{r}'(\tau)\right\| d\tau\,. \tag{2}$$

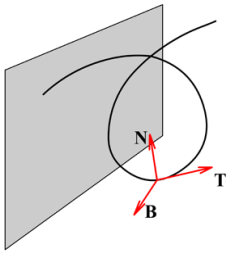

Figure 8. TNB frame

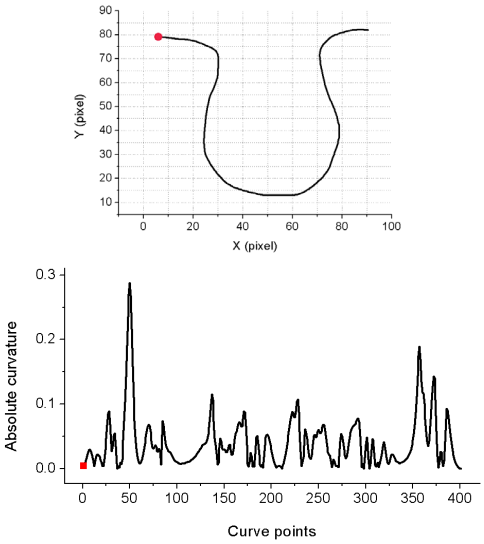

Figure 9. 2D curve and absolute curvature

This equation can be rewritten as the curve as $r(s)=r(t(s))$, based on the arc length description, curvature κ and torsion τ can be defined by Frenet-Serret formulas [9]:

$$\begin{bmatrix}\mathbf{T}'\\ \mathbf{N}'\\ \mathbf{B}'\end{bmatrix}=\begin{bmatrix}0 & \kappa & 0\\ -\kappa & 0 & \tau\\ 0 & -\tau & 0\end{bmatrix}\begin{bmatrix}\mathbf{T}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix}, \tag{3}$$

where $\mathbf{T}$, $\mathbf{N}$, $\mathbf{B}$ denotes unit tangent, normal and bi-normal respectively. The apostrophe means derivative with arc length. Relationship between each vector can also be described as Frenet-Serret frame (Also known as TNB frame), shows in Fig.8:

$$\mathbf{T}=\frac{d\mathbf{r}}{ds},\quad \mathbf{N}=\frac{\frac{d\mathbf{T}}{ds}}{\left\|\frac{d\mathbf{T}}{ds}\right\|},\quad \mathbf{B}=\mathbf{T}\times\mathbf{N}. \tag{4}$$

B. 2D Open Curve Matching

The equations described in Section III.A are applicable for any dimension. As a special case, 2D open curve can be calculated by (3) and (4) without torsion and bi-normal description. After defining the start point of a 2D curve, the curvature parameter sequence can be realized as a string which denotes the specific 2D curve in the feature space. As shown in the Fig.9, the graph on the top is the geometrical distribution of a simple curve starts from the round dot and the plot below the curve shows the absolute values of the curvature.

The 2D open curve matching can be represented by following algorithm:

Given two curves $\mathbf{C}_1=\{(x_{11},y_{11}),(x_{12},y_{12}),\ldots,(x_{1n},y_{1n})\}$ and $\mathbf{C}_2=\{(x_{21},y_{21}),(x_{22},y_{22}),\ldots,(x_{2m},y_{2m})\}$ in an Euclidean space, each one can be denoted by its curvature:$\mathbf{C}_1=\{\kappa_{11},\kappa_{12},\ldots,\kappa_{1n}\}$ and $\mathbf{C}_2=\{\kappa_{21},\kappa_{22},\ldots,\kappa_{2n}\}$. Initialize a distance matrix $\mathbf{D}(i,j)$ with $(n+1)\times(m+1)$ elements, this string edit operation computes each element of $\mathbf{D}(i,j)$ by the edit algorithm. Shown as (5), (6) and (7)

$$\mathbf{D}(i,j) = d(\mathbf{C}'_1, \mathbf{C}'_2), \quad (5)$$

$$\mathbf{C}'_1 = \{\kappa_{11}, \kappa_{12}, \ldots, \kappa_{1i}\}, \quad (6)$$

$$\mathbf{C}'_2 = \{\kappa_{21}, \kappa_{22}, \ldots, \kappa_{2j}\}. \quad (7)$$

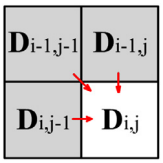

Figure.10 "String Edit" operation

The lower right corner value of the matrix records the difference of the two curves. Each element of this matrix is calculated based on a previous iteration operation. As shown in the Fig.10, there are three probable predecessors, each one denoting one specific string operation: **D**(*i*-1,*j*-1)-substitute, **D**(*i*-1,*j*)-delete, and **D**(*i*,*j*-1)-insert. In addition, it can also be used to trace the changes from one curve into another one based on the three string edit operations. The complexity of this algorithm is of *O*(nm).

Pseudocode for implementing the algorithm in the experiment is shown below:

Pseudocode stringEdit ($\mathbf{C}_1$,$\mathbf{C}_2$)
START:
//Initialization
int n = length of $\mathbf{C}_1$;
int m = length of $\mathbf{C}_2$;
array D[n+1][m+1] = 0;
//Initialization first row and column of D
For i = 1 to n
    D[i][0] = D[i-1][0] + c(delete$_{i0}$);
For j = 1 to n
    D [0][j] = D[0][j-1] + c(insert$_{0j}$);
//Distance iterative calculation
For i = 1 to n
    For j = 1 to m
    {
        m1 = D[i-1][j-1] + c(substitute$_{ij}$);
        m2 = D[i-1][j] + c(delete$_{ij}$);
        m3 = D [i][j-1] + c(insert$_{ij}$);
        D [i][j] = min (m1, m2 , m3);
    }
//Output result
float difference = D [n][m]
END Pseudocode

In the pseudocode, a function of operation cost *c*(*operation*) is introduced. It defines the cost of the three edit operations. The rules can be changed based on different applications. For this 2D curve matching problem, the cost functions are defined as:

$$c(delete_{ij}) = c(insert_{ij}) = 0.1, \quad c(subsitude_{ij}) = |\kappa_i - \kappa_j|, \quad (6)$$

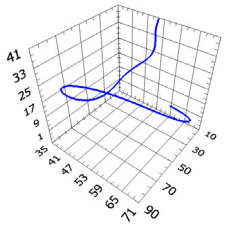

Figure.11. 3D curve example

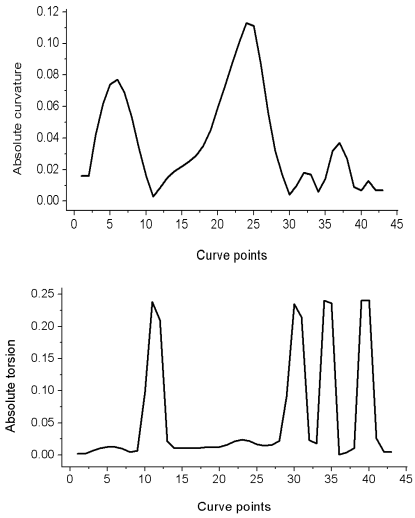

Figure.12. Absolute curvature and torsion

C. *3D Open Curve Matching*

"String Edit" operations have been introduced into this research for 3D curve matching. In Euclidian space, a 3D curve can be described by absolute coordinate values or by curvatures and torsions along the curves. An example of 3D curve and its corresponding curvatures and torsions is shown in Fig.11 and 12.

For example, given two 3D curves:

$\mathbf{C}_1=\{(x_{11},y_{11},z_{11}),(x_{12},y_{12},z_{12}),\ldots,(x_{1n},y_{1n},z_{1n})\}$

and

$\mathbf{C}_2=\{(x_{21},y_{21},z_{21}),(x_{22},y_{22},z_{22}),\ldots,(x_{2m},y_{2m},z_{2m})\}$

in the Euclidian space, the curves can be described by the curvature and torsion vector series, denoted as

$\mathbf{C}_1=\{(\kappa_{11},\tau_{11}),(\kappa_{12},\tau_{12}),\ldots,(\kappa_{1n},\tau_{1n})\}$

and

$\mathbf{C}_2=\{(\kappa_{21},\tau_{21}),(\kappa_{22},\tau_{22}),\ldots,(\kappa_{2m},\tau_{2m})\}$.

These two strings are used to calculate the distance matrix. This differs from the 2D approach as costs functions are changed to reflect the curvature and torsion pairs' vector nature in the forms of:

$$\begin{aligned} c(delete_{ij}) &= \|(\kappa_i, \tau_i)\|; \\ c(insert_{ij}) &= \|(\kappa_j, \tau_j)\|; \\ c(subsitude_{ij}) &= \|(\kappa_i, \tau_i) - (\kappa_j, \tau_j)\| \end{aligned}, \quad (8)$$

D. *High-dimensional String Edit Operation Extension*

The string edit operation can be extended to other high dimensional pattern matching applications. The only prerequisite is that a pattern to be described (if viable) as a series of parameters can be composed into a "vector string". The distance matrix can then be modified to suit for the specific high dimensional parameters vectors through redefining the cost functions as illustrated in (8).

IV. Experiment Design and Result Analysis

The volume-based event detection algorithms introduced in Section 2 and 3 are developed and tested on simulation platforms MATLAB and LabVIEW running on a PC equipped with an AMD Athlon 2.62GHz CPU and 2G RAM. Experimental video samples were downloaded from the Weizmann video library database published by Lena Gorelick [10]. Event curve templates were manually plotted for exploring 3D open curve matching techniques on a frame-by-frame basis. Fig.13 shows various event templates and related geometry features, which are represented by ordered feature points from key parts of the human stick model.

To assess the effectiveness and efficiency of the sting-based 3D curve matching approach, the experiments for matching have been firstly tested on a number of artificial 3D curve templates as shown in Fig.14. The curve sets contain four similar curves and four random curves. The matching results between each curve pair are listed in the Table.1. The results clearly indicate that the curves 1 to 4 are identical, which can be verified by visual inspection. Time consumptions listed in the table reflect the relationship between the curve length and algorithm Time-and-Space-Complexity.

It was concluded during the preliminary experiments that the 3D curve matching approach possesses the potential for predicting human movements or actions by simply tracking the head trajectory. Events such as waving, walking, jumping and bending can all be represented by distinctive head trajectories.

By careful adjusting the threshold, an approximate 66.8% of template autocorrelation will see the successful separation of the positively and negatively matched curves. In the Table.2, event patterns listed in the Fig.13 are compared with the "wave two hands" template. The five feature curves (head, left and right hands/feet), all match the event template separately. After template matching operation, if the five curves match the template, the event can be recognized from the pattern video

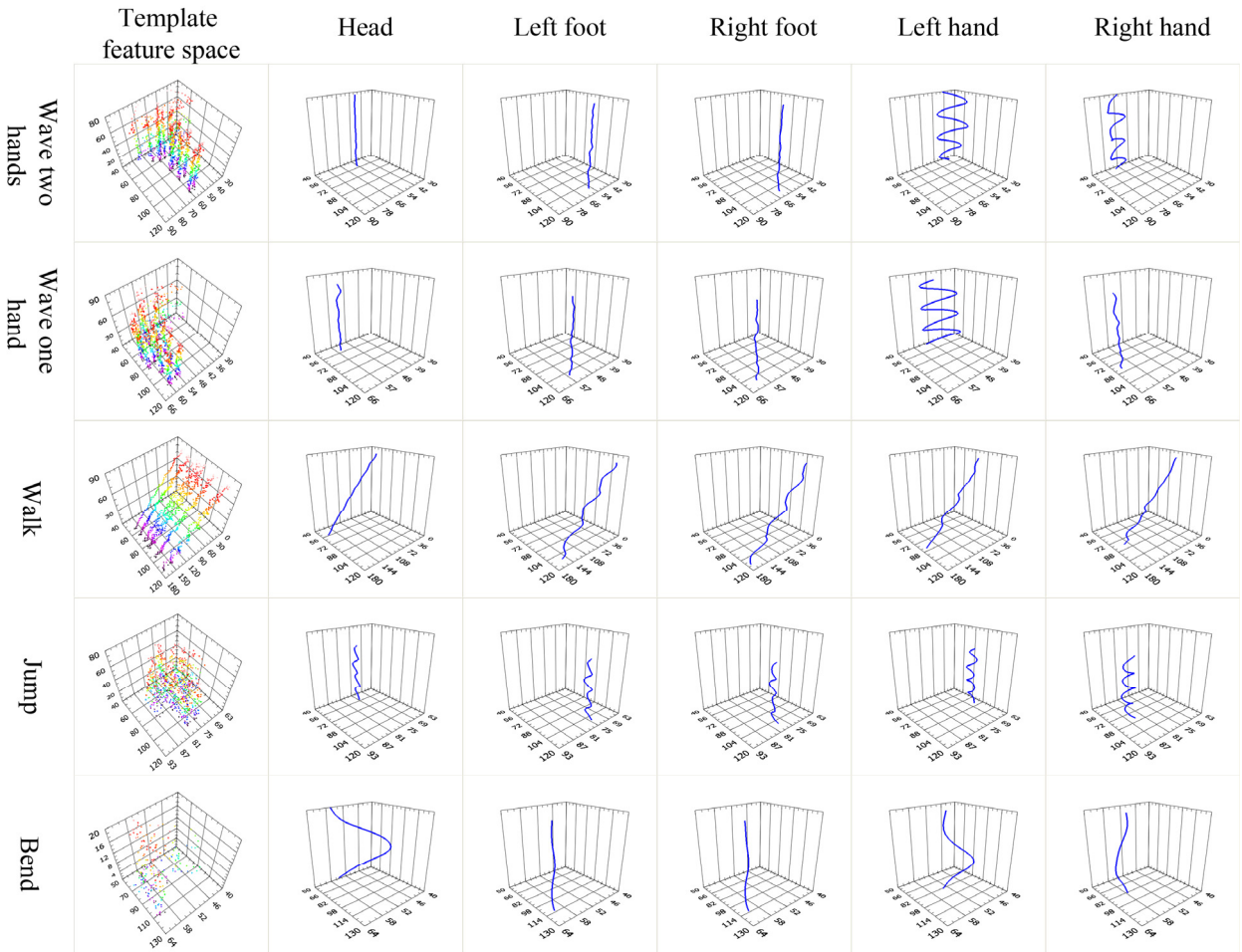

Figure 13. Volume-based event feature points and curves on selected Weizmann video library data

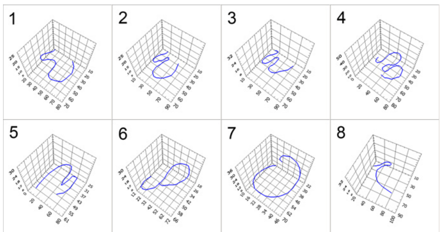

Figure 14. Artificial 3D open curves

8 (Time consumption)								110
8 (Similarity)								1
7 (Time consumption)							154	102
7 (Similarity)							1	0.40
6 (Time consumption)						92	131	99
6 (Similarity)						1	0.33	0.34
5 (Time consumption)					112	109	140	110
5 (Similarity)					1	0.12	0.43	0.10
4 (Time consumption)				84	93	87	119	90
4 (Similarity)				1	0.45	0.09	0.52	0.23
3 (Time consumption)			103	90	109	94	137	104
3 (Similarity)			1	0.79	0.31	0.29	0.15	0.32
2 (Time consumption)		68	79	71	95	83	103	84
2 (Similarity)		1	0.96	0.81	0.22	0.37	0.17	0.48
1 (Time consumption)	98	87	96	89	104	95	137	106
1 (Similarity)	1	0.92	0.89	0.87	0.52	0.61	0.23	0.45
	1	2	3	4	5	6	7	8

Table 1. "String Edit" 3D open curve matching

	Head	**Left foot**	**Right foot**	**Left hand**	**Right hand**
Wave two hand	90.3%	82.7%	91.1%	87.1%	82.9%
Wave one hand	92.7%	81.5%	82.3%	78.2%	20.2%
Walk	12.3%	7.8%	10.6%	9.4%	8.1&
Jump	60.4%	54.8%	45.6%	70.7%	61.5%
Bend	5.2%	86.2%	89.6	33.7%	41.5%

Table 2. "Wave two hand" template compared with pattern curves on selected Weizmann video library data

V. CONCLUSION AND FUTURE WORK

After the human stick-model generated, the event detection process is transformed into a series of operations on feature-based entities within the 3D spatio-temporal volume space. For example, human gestures and actions can be transformed into spatial curves with distinctive analytical geometrical characteristics and probabilistic distributions. Or in other words, event detection problems can be "simplified" into 3D curve analysis tasks. In the early trials, the geometrical analysis and template matching for the curves have been treated as two inter-related steps. As the result demonstrated, event curves - especially the head curves - provide key information on the primary human motions. Detailed analysis has been performed regarding the 3D curve matching algorithms developed from the so-called "String Edit" method denoted by the "cost function" and "distance matrix", which is an extension of the popular 2D contour matching solution for script matching. Test results show that the "String Edit" approach can distinguish different curves by matching the distance and the curvature-torsion parameters, which can be readily introduced into other high-dimension vector-based matching problems.

There are a number of promising directions for future works based on this project. Firstly, it is understood that the curvature-torsion-pairs-based event (curve) matching is a pattern analysis process which might need an automated learning system as a common practice depends on application scenarios. Future work will assess and benchmark the proposed matching technique for its automation potentials.

As evident in the experiments, the STV volume data structure adopted in this research introduces substantial work load and time-consumption to the computing platform. Although this work is not focusing on the real-time performance of the devised event detection method, operation efficiency still plays an important role in many real-world applications. One of the potential methods for solving this problem is through employing hardware acceleration on modern PC equipment. For example, to employ the Graphics Processing Unit (GPU) for accelerating the computation [11], and using OpenCL for more efficient CPU/GPU workload distribution.

REFERENCES

[1] E. Aldelson, and J. R. Bergen, "Spatiotemporal Energy Models for the Perception of Motion," *Journal Optical Society of America,* vol. 2, pp. 284-299, 1985.

[2] G. Rogez, C. Orrite-Urunuela, and J. Martinez-del-Rincon, "A Spatio-temporal 2D-models Framework for Human Pose Recovery in Monocular Sequences," *Pattern Recognition,* vol. 41, no. 9, pp. 2926-2944, 2008.

[3] Y. Li, and G. R. Arce, "A Maximum Likelihood Approach to Least Absolute Deviation Regression," *EURASIP Journal on Applied Signal Processing,* vol. 2004, no. 12, pp. 1762-1769, 2004.

[4] J. O. Berger, "Certain Standard Loss Functions," *Statistical Decision Theory and Bayesian Analysis (2nd ed.)*, p. 60, New York: Springer-Verlag, 1985.

[5] J. Parr, and C. Philips, *Sugnals, Systems and Transforms (2nd ed.)*, New Jersey: Prentice Hall, 1999.

[6] H. Bunke, and U. Buhler, "Applications of Approximate String Matching to 2D Shape Recognition," *Pattern Recognition,* vol. 26, no. 12, pp. 1797-1812, 1993.

[7] R. A. Wanger, and M. J. Fischer, "The String-to-String Correction Problem," *Journal of the ACM,* vol. 21, pp. 168-173, 1974.

[8] K. Arbter, "Application of Affine-Invariant Fourier Descriptors to Recognition of 3-D Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, pp. 640-647, 1990.

[9] H. C. Crenshaw, and L. E. Keshet, "Orientation by Helical Motion II. Changing the Direction of the Axis ofmotion," *Bulletin of Mathematical Biology,* vol. 55, no. 1, pp. 213-230, 1993.

[10] L. Gorelick, and M. Blank, "Actions as Space-Time Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 29, no. 12, pp. 2247-2253, 2007.

[11] F. Porikli, "Constant time O(1) Bilateral Filtering," in CVPR, 2008.